5

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-066954
(43)Date of publication of application : 19.03.1993

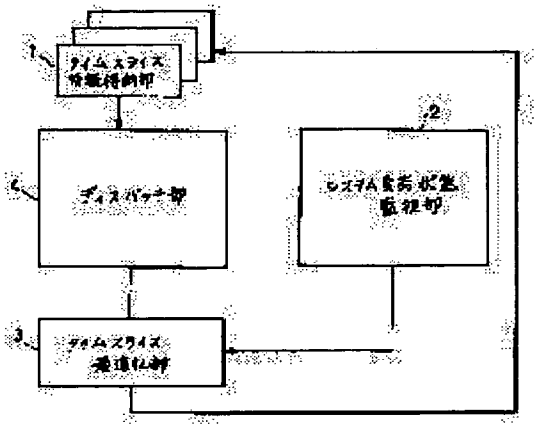
(51)Int.Cl. G06F 9/46
G06F 15/00

(21)Application number : 03-252909
(22)Date of filing : 05.09.1991

(71)Applicant : KOBE NIPPON DENKI SOFTWARE KK
(72)Inventor : SHIBANO RYOICHI

(54) TIME SLICE OPTIMIZATION SYSTEM

(57)Abstract:
PURPOSE: To facilitate the effective operation of a time slice optimization system by adjusting automatically the time slice value of each program in accordance with the system state.
CONSTITUTION: A time slice information storing part 1 stores the time slice information on each program. A dispatching part 4 dispatches a CPU to each program based on the time slice value stored in the part 1. A system load state monitoring part 2 monitors the load state of the system. Then a time slice optimizing part 3 optimizes the time slice value of each program stored in the part 1 based on the information that is carried out through the part 4 at a fixed interval and acquired through the part 2.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平5－66954

(43)公開日　平成５年(1993)３月19日

(51)Int.Cl.[5]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	9/46	３４０　Ｅ	8120－5B		
		Ｄ	8120－5B		
	15/00	３１０　Ｋ	8219－5L		

審査請求　未請求　請求項の数１(全　９　頁)

(21)出願番号　特願平3－252909

(22)出願日　平成３年(1991)９月５日

(71)出願人　000192545
神戸日本電気ソフトウエア株式会社
兵庫県神戸市西区高塚台５丁目３番１号

(72)発明者　芝野　良一
兵庫県神戸市西区高塚台５丁目３番１号
神戸日本電気ソフトウエア株式会社内

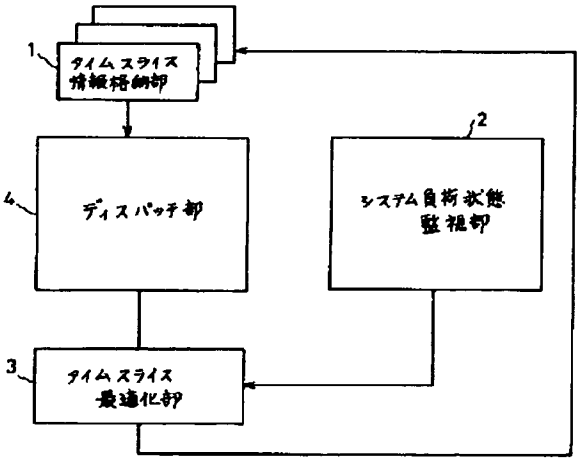
(74)代理人　弁理士　山川　政樹

(54)【発明の名称】　タイムスライス最適化方式

(57)【要約】

【目的】　各プログラムのタイムスライス値をシステムの状態に合わせて自動的に調整を行い、容易に効率的なシステム運用が行えるようにする。

【構成】　タイムスライス情報格納部１により各プログラムのタイムスライス情報を格納し、ディスパッチ部４でこのタイムスライス情報格納部１に格納されているタイムスライス値を基に各プログラムにＣＰＵをディスパッチし、システム負荷状態監視部２でシステムの負荷状態を監視し、タイムスライス最適化部３によりディスパッチ部４から一定間隔で実行されシステム負荷状態監視部２から得た情報を基にタイムスライス情報格納部１に格納している各プログラムのタイムスライス値を最適化する構成にした。



1

【特許請求の範囲】

【請求項１】　各プログラム毎に異なるタイムスライス値を用いてＣＰＵディスパッチを行うマルチプログラミングシステムにおいて、現在実行している各プログラム毎にそのプログラムに関するタイムスライス情報を格納するタイムスライス情報格納部と、一定間隔でシステムの負荷状況を監視しているシステム負荷状態監視部と、一定期間毎に前記システム負荷状態監視部より取得したその期間のシステム負荷状態より前記タイムスライス情報格納部に格納した各プログラムのタイムスライス値を最適化するタイムスライス最適化部と、このタイムスライス最適化部で最適化して前記タイムスライス情報格納部に格納しているタイムスライス値で前記ＣＰＵをプログラムにディスパッチするディスパッチ部を設け、システムの状態に合わせて自動的にタイムスライス値を調整するようにしたことを特徴とするタイムスライス最適化方式。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は各プログラム毎に異なるタイムスライスを用いてＣＰＵディスパッチを行うマルチプログラミングシステムにおいて、システムの負荷状態に合わせて自動的にタイムスライス値の調整をすることができるタイムスライス最適化方式に関するものである。

【０００２】

【従来の技術】従来、マルチプログラミングシステムでは、特定のプログラムが長時間ＣＰＵを占有することを防ぐためにタイムスライスを用いたＣＰＵディスパッチを行っていた。このタイムスライスはプログラム毎または複数のプログラムからなるプログラムグループ毎に設定される。そして、一般に計算処理を主体とするプログラムの場合はディスパッチされたＣＰＵをタイムスライスを使いきって放棄することが多く、逆にインタラクティブ処理や入出力処理が主体のプログラムの場合はディスパッチされたＣＰＵを入出力の完了待ちやその他のイベント待ち等の理由でタイムスライスを使いきらずに放棄することが多い。また、ＣＰＵの使用が放棄された時点でディスパッチング処理が実行されシステムのオーバヘッドの一部となるため、システムの使用効率をより高めるためにはタイムスライスの値を大きくし単位時間当たりのディスパッチング処理の実行回数を減らすことが有用である。

【０００３】しかし、各プログラムのタイムスライス値を大きくすると、システムの負荷が軽い場合は特に問題にはならないが、システムの負荷が高くなるとインタラクティブ処理を主体とするプログラムの応答性が極端に遅くなってしまう。したがって、システムの負荷が軽い間は長めのタイムスライス値が有効であり、負荷が高くなると短めのタイムスライス値が有効となる。

2

【０００４】

【発明が解決しようとする課題】上述した従来の方式では、各プログラムに設定するタイムスライスは固定であり、システムの状態に合わせて自動的に変更することはできないという課題があった。

【０００５】

【課題を解決するための手段】本発明のタイムスライス最適化方式は、各プログラム毎に異なるタイムスライス値を用いてＣＰＵディスパッチを行うマルチプログラミングシステムにおいて、現在実行している各プログラム毎にそのプログラムに関するタイムスライス情報を格納するタイムスライス情報格納部と、一定間隔でシステムの負荷状況を監視しているシステム負荷状態監視部と、一定期間毎に上記システム負荷状態監視部より取得したその期間のシステム負荷状態より上記タイムスライス情報格納部に格納した各プログラムのタイムスライス値を最適化するタイムスライス最適化部と、このタイムスライス最適化部で最適化して上記タイムスライス情報格納部に格納しているタイムスライス値で上記ＣＰＵをプログラムにディスパッチするディスパッチ部を設け、システムの状態に合わせて自動的にタイムスライス値を調整するようにしたものである。

【０００６】

【作用】本発明においては、一定期間毎にその間のシステム負荷状態により各プログラムのタイムスライス値を見直し、最適なシステム運用が行えるように自動的にタイムスライス値の調整を行うようにする。

【０００７】

【実施例】図１は本発明の一実施例を示すブロック図である。この図１において、１は現在実行している各プログラム毎にそのプログラムに関するタイムスライス情報を格納するタイムスライス情報格納部、２は一定間隔でシステムの負荷状況を監視しているシステム負荷状態監視部、３は一定期間毎にシステム負荷状態監視部２より取得したその期間のシステム負荷状態よりタイムスライス情報格納部１に格納した各プログラムのタイムスライス値を最適化するタイムスライス最適化部、４はこのタイムスライス最適化部３で最適化してタイムスライス情報格納部１に格納しているタイムスライス値でＣＰＵをプログラムにディスパッチするディスパッチ部である。

【０００８】ここで、上記システム負荷状態監視部２は周期的にシステムの負荷状況を監視しているシステム負荷状態監視部であり、タイムスライス最適化部３は一定期間毎にシステム負荷状態監視部２より得る現在のシステムの負荷状況によって各プログラムに設定したタイムスライス値に対する調整値を計算しタイムスライス情報格納部１に格納された各プログラムのタイムスライス値を調整するタイムスライス最適化部である。また、ディスパッチ部４は次に実行すべきプログラムを決定しそのプログラムに対しタイムスライス情報格納部１に格納さ

れたタイムスライス値でＣＰＵをディスパッチするとともに一定期間毎にタイムスライス最適化部３を実行するディスパッチ部である。そして、システムの状態に合わせて自動的にタイムスライス値を調整するように構成されている。

【０００９】図２は図１におけるタイムスライス情報格納部１の構成例を示す説明図である。この図２において、１１はそのプログラムに最初に設定されたタイムスライス初期値を示し、１２は現在のタイムスライス値、１３は取り得ることのできるタイムスライス値の最大値、１４はタイムスライス値の最小値を示す。そして、これらの情報は対応するプログラムが実行可能になった時点で新しく用意され値が格納される。その場合、現在のタイムスライス値１２の値はタイムスライス初期値１１と同じ値が格納される。

【００１０】図３は図１におけるディスパッチ部４のアルゴリズムの例を示すフローチャートである。まずシステム起動時にディスパッチ部４がタイムスライス最適化部３を実行する間隔を設定する（ステップ１０１）。つぎに、実行すべきプログラムを決定し、そのプログラムのタイムスライス情報を格納しているタイムスライス情報格納部１を参照し、図２に示す現在のタイムスライス値１２だけそのプログラムにＣＰＵをディスパッチする（ステップ１０２～１０４）。そして、そのプログラムがＣＰＵの使用を放棄すると、ディスパッチ部４は以前タイムスライス最適化部３を実行してから設定した期間が過ぎたかどうかを調べる（ステップ１０５，１０６）。過ぎていた場合にはタイムスライス最適化部３を実行し、次にタイムスライス最適化部３を実行するまでの期間を設定する（ステップ１０７，１０８）。その後、ディスパッチ部４は次に実行するプログラムの選定に戻る。

【００１１】図４は図１におけるタイムスライス最適化部３のアルゴリズムの例を示すフローチャートである。まず、タイムスライス最適化部３はディスパッチ部４により一定期間毎に実行される。そして、タイムスライス最適化部３はシステム負荷状態監視部２より前回調べてから今までのシステムの負荷状況を取得する。この例ではシステムの負荷状況としてその期間中のアイドルＣＰＵ比率を取得している（ステップ２０１）。つぎに、このシステム負荷状況よりタイムスライス値の調整を行うべきか否か、または調整する場合はその調整値をどのくらいにするかを決定する。この例の場合はアイドルＣＰＵ比率が２５％以下の場合はシステムの負荷が高いとみなし（ステップ２０２）、次回からのタイムスライス値を減らすために調整値としてマイナスの値（－△）を設定する（ステップ２０３）。逆にアイドルＣＰＵ比率が７５％以上の場合（ステップ２０４）はシステムの負荷が軽いとみなし、次回からのタイムスライス値を増やすために調整値としてプラスの値（＋△）を設定する（ステップ２０５）。ここで、アイドルＣＰＵ比率が２５％以下でも８５％以上でもない場合には、現在のタイムスライス値が妥当であるとみなし、タイムスライス値の調整は行わない。

【００１２】つぎに、各実行中のプログラムに対応するタイムスライス情報格納部１を順番に調べ（ステップ２０６）、タイムスライス情報格納部１内の現在のタイムスライス値１２の値に上記調整値を加える（ステップ２０７）。ただし、加算した後の値が上記タイムスライス情報格納部１内のタイムスライス値の最大値１３より大きくなった場合は（ステップ２０８）、現在のタイムスライス値１２の値をタイムスライス値の最大値１３にする（ステップ２０９）。これはタイムスライス値が無限に大きくなるのを防ぐためである。また、加算した後の値が上記タイムスライス情報格納部１内のタイムスライス値の最小値１４より小さくなった場合には（ステップ２１０）、現在のタイムスライス値１２の値をタイムスライス値の最小値１４にする（ステップ２１１）。これはタイムスライス値が「０」以下になることを防ぐためである。

【００１３】そして、上記調整値の決め方は図４の例の方法の他に図５に示すような表を用いて行うこともできる。図５は図１におけるタイムスライス最適化部３で使用する調整値決定用の表の一例を示す説明図である。この場合はシステム負荷状態監視部２より得たアイドルＣＰＵ比率で表の条件部を検索し、合致した条件部の横の調整値をタイムスライスの調整値とするとともに、その値が「０」の場合は調整の必要がないものとする。

【００１４】図６は図１におけるシステム負荷状態監視部２の構成例を示すブロック図である。システム負荷状態監視部２は、一定時間間隔で割り込みを発生させるタイマ部２１と、このタイマ部２１の発生する割り込み毎にシステムの負荷状態を調べるシステム状態調査部２２と、このシステム状態調査部２２の結果を記録保存するシステム状態記録部２４と、タイムスライス最適化部３からの状態問い合わせに対してシステム状態記録部２４の内容より現在のシステムの負荷状態を答えるシステム状態回答部２３から構成されている。

【００１５】図７はこの図６におけるシステム状態調査部２２のアルゴリズムの例を示すフローチャートである。システム状態調査部２２は、起動時にまず、システム状態記録部２４内の各値を「０」にクリアし（ステップ３０１）、つぎに、システム状態を監視する間隔を示す割り込み間隔をタイマ部２１に設定する（ステップ３０２）。タイマ部２１からの割り込みが発生する毎にＣＰＵの状態を調べ（ステップ３０３，３０４）、ＣＰＵが使用中であれば（ステップ３０５）、システム状態記録部２４内のＣＰＵビジー回数に「１」足す（システム３０６）。使用中で無い場合には同じくシステム状態記録部２４内のＣＰＵアイドル回数に「１」足す（ステッ

5

ブ３０７）。その後再度割り込み待ちに戻る。

【００１６】図８は図６におけるシステム状態回答部２３のアルゴリズムの例を示すフローチャートである。まず、システム状態回答部２３は、タイムスライス最適化部３より実行され、システム状態記録部２４の内容を調べシステムの負荷状態を回答する。この例では、まず、システムの負荷状態として図８の最初のステップ４０１で計算されるアイドルＣＰＵ比率を求める。つぎに、システム状態記録部２４内の各値を「０」にクリアする（ステップ４０２）。最後に上記で求めたアイドルＣＰＵ比率を問い合わせ元であるタイムスライス最適化部３に回答する（ステップ４０３）。

【００１７】

【発明の効果】以上説明したように本発明は、一定期間毎にその間のシステム負荷状態により各プログラムのタイムスライス値を見直し、最適なシステム運用が行えるように自動的にタイムスライス値の調整を行うようにしたので、一定期間毎にシステムの負荷状況により自動的に各プログラムのタイムスライス値を調整することができるため、容易に効率の良い最適なシステム運用ができるという効果がある。

6

＊【図面の簡単な説明】

【図１】本発明の一実施例を示すブロック図である。

【図２】図１におけるタイムスライス情報格納部の構成例を示す説明図である。

【図３】図１におけるディスパッチ部のアルゴリズムの例を示すフローチャートである。

【図４】図１におけるタイムスライス最適化部のアルゴリズムの例を示すフローチャートである。

【図５】図１におけるタイムスライス最適化部で使用する調整値決定用の表の一例を示す説明図である。

【図６】図１におけるシステム負荷状態監視部の構成例を示すブロック図である。

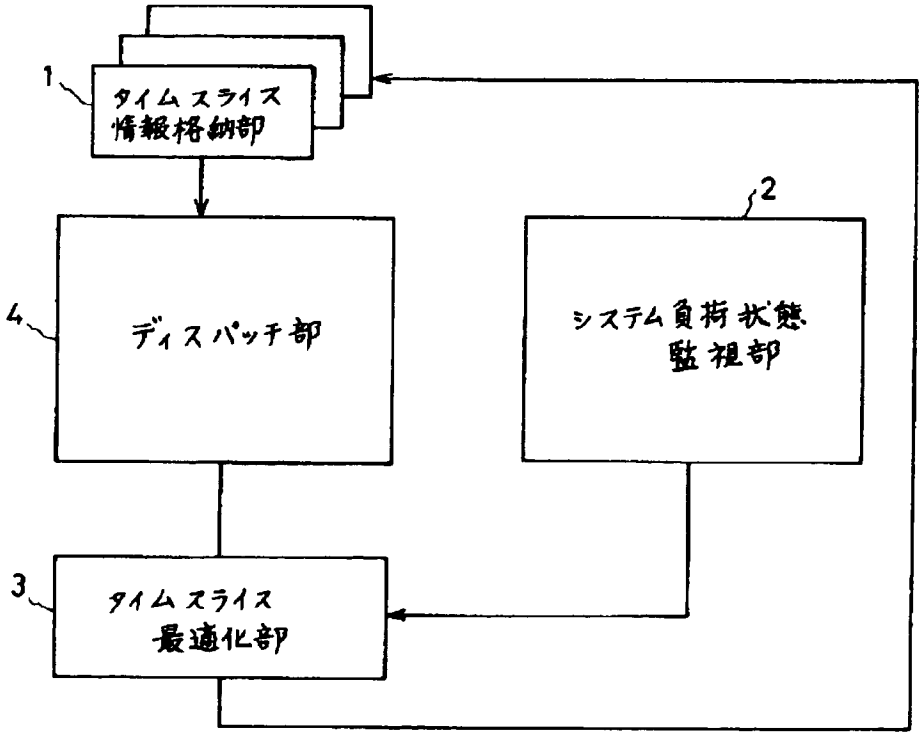
【図７】図６におけるシステム状態調査部のアルゴリズムの例を示すフローチャートである。

【図８】図６におけるシステム状態回答部のアルゴリズムの例を示すフローチャートである。

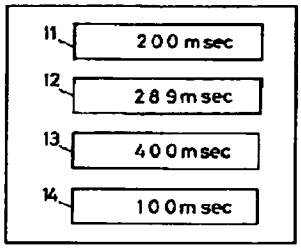
【符号の説明】

１　タイムスライス情報格納部
２　システム負荷状態監視部
３　タイムスライス最適化部
４　ディスパッチ部

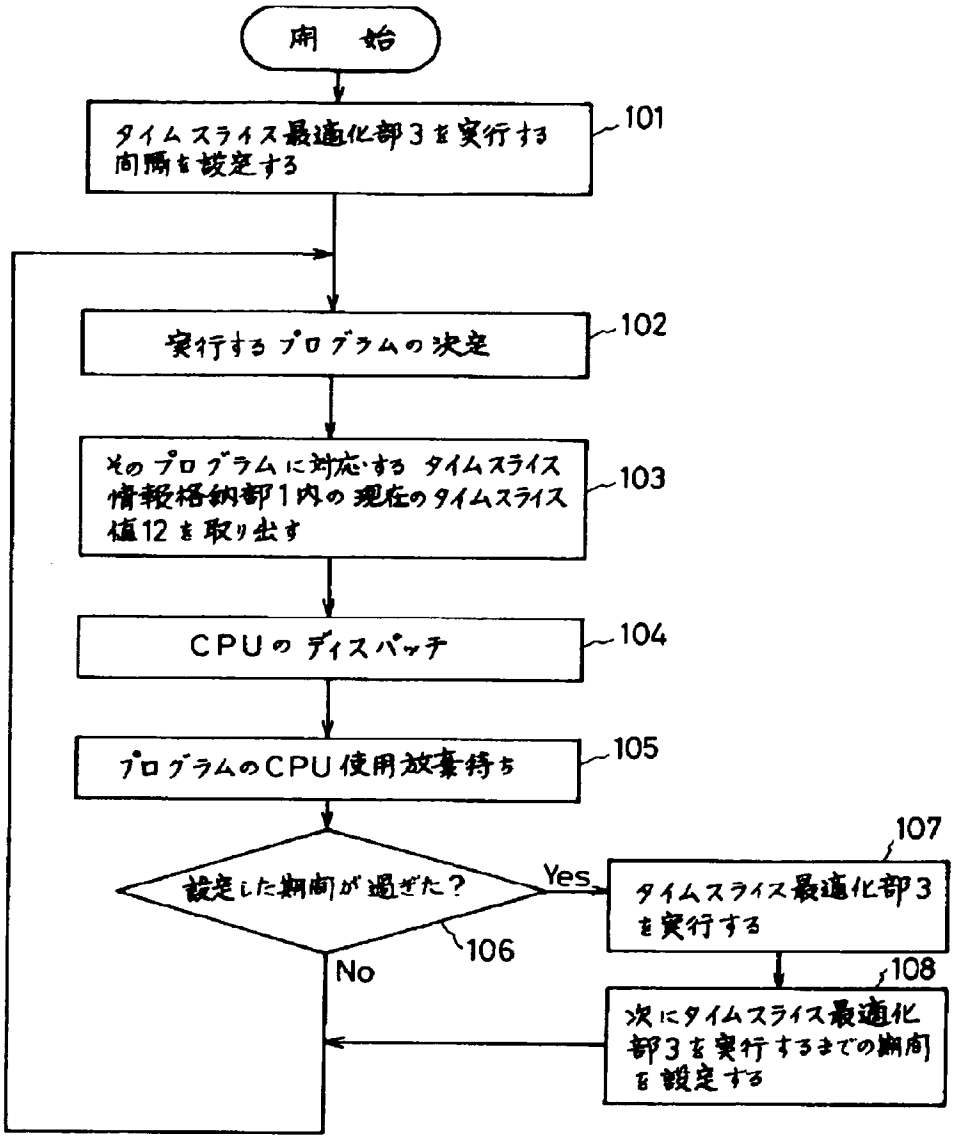
【図１】



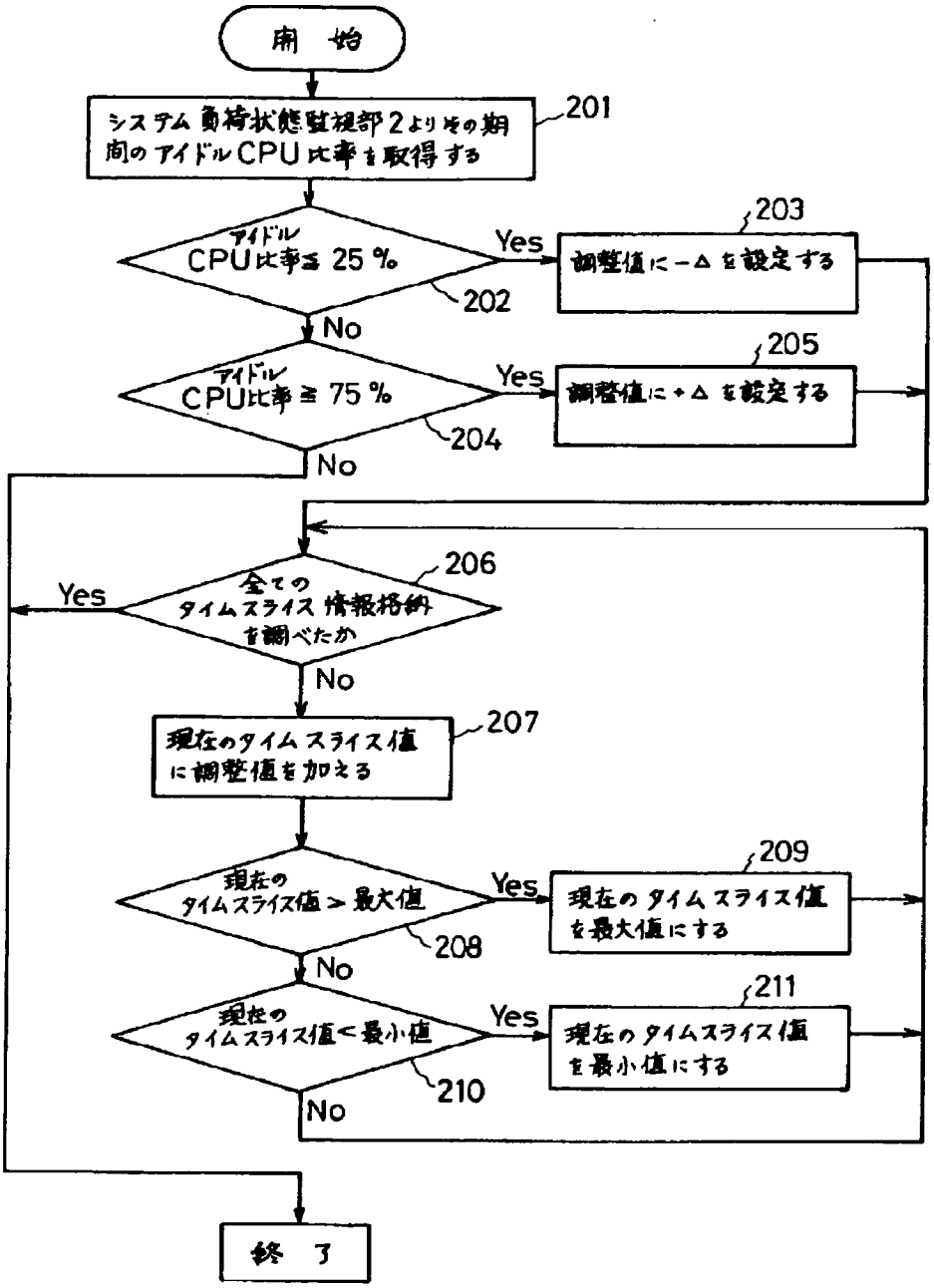
【図２】



【図3】


開　始
タイムスライス最適化部3を実行する間隔を設定する
101
実行するプログラムの決定
102
そのプログラムに対応するタイムスライス情報格納部1内の現在のタイムスライス値12を取り出す
103
CPUのディスパッチ
104
プログラムのCPU使用放棄待ち
105
設定した期間が過ぎた？
106
Yes
No
107
タイムスライス最適化部3を実行する
108
次にタイムスライス最適化部3を実行するまでの期間を設定する

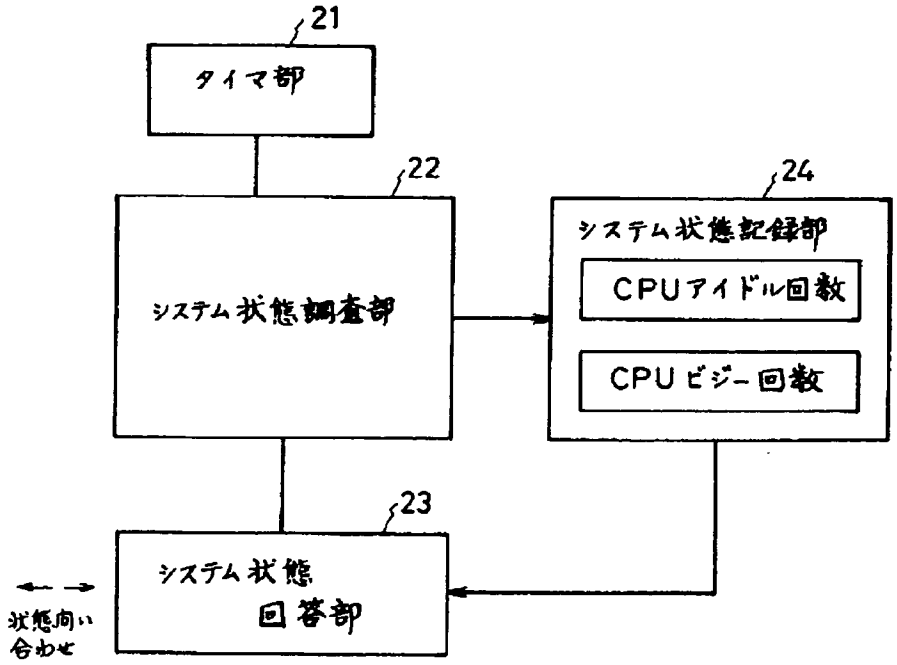
【図4】


開始
システム負荷状態監視部2よりその期間のアイドルCPU比率を取得する
201
アイドルCPU比率≧25%
202
Yes
203
調整値に－Δを設定する
No
アイドルCPU比率≧75%
204
Yes
205
調整値に＋Δを設定する
No
全てのタイムスライス情報格納を調べたか
206
Yes
No
現在のタイムスライス値に調整値を加える
207
現在のタイムスライス値＞最大値
208
Yes
209
現在のタイムスライス値を最大値にする
No
現在のタイムスライス値＜最小値
210
Yes
211
現在のタイムスライス値を最小値にする
No
終了

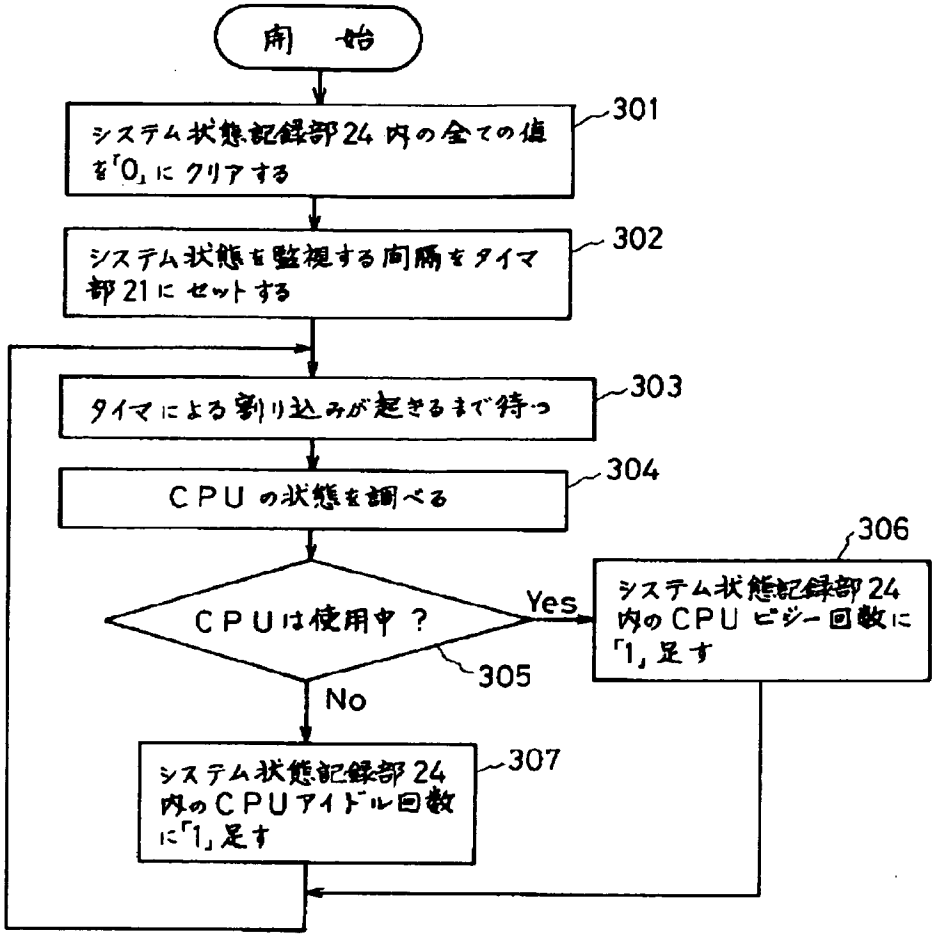
【図5】

条件部	調整値
0％ ～ 10％	-10msec
11％ ～ 25％	-5msec
26％ ～ 75％	0msec
76％ ～ 90％	+5msec
91％ ～ 100％	+10msec

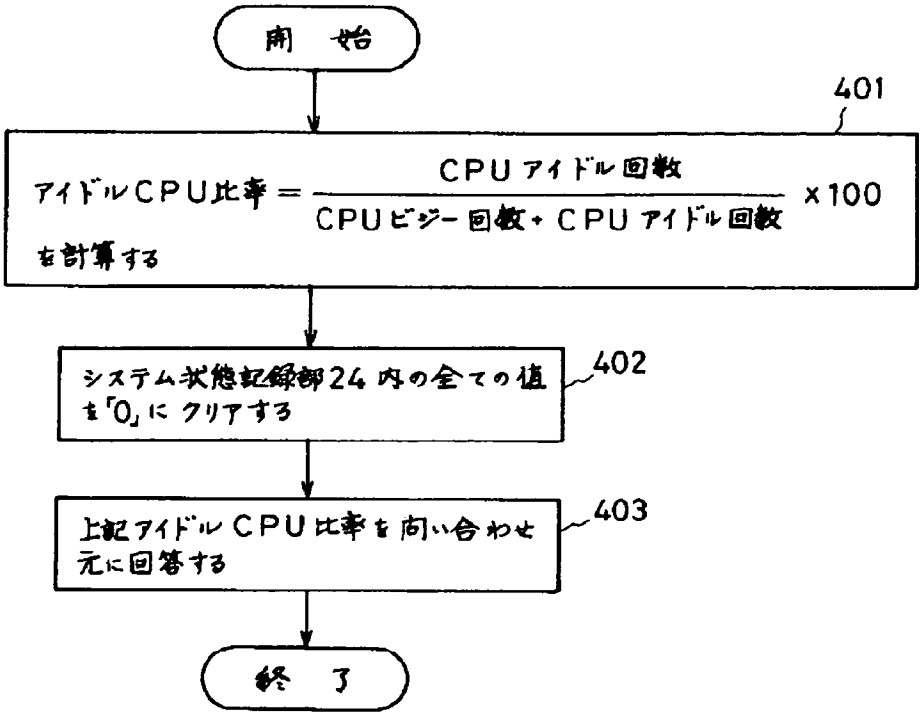
【図6】



【図7】


開　始
システム状態記録部24内の全ての値を「0」にクリアする
301
システム状態を監視する周期をタイマ部21にセットする
302
タイマによる割り込みが起きるまで待つ
303
CPUの状態を調べる
304
CPUは使用中？
Yes
No
305
システム状態記録部24内のCPUビジー回数に「1」足す
306
システム状態記録部24内のCPUアイドル回数に「1」足す
307

【図8】


開　始
401
アイドルCPU比率＝CPUアイドル回数 / CPUビジー回数＋CPUアイドル回数 ×100
を計算する
システム状態記録部24内の全ての値を「0」にクリアする
402
上記アイドルCPU比率を問い合わせ元に回答する
403
終　了